

Documentação Desafio

Rafaela de Moraes Papale

Requisitos do Sistema

- » **Ambiente de execução:** Node.js.
- » **Framework:** React.
- » **Banco de dados:** MySQL.
- » **Código do projeto:** <https://github.com/RafaelaPapale/processo-seletivo-nouvenn>

1º Parte – Banco de dados

Primeiramente, foi criado um script .sql que cria um banco de dados chamado “nouvenn”. O banco de dados MySQL foi escolhido por se tratar de um sistema em que os dados são relacionais.

Antes de rodar o back-end, é preciso executar o arquivo .sql.

2º Parte – Back-end

Após a criação do banco de dados, é possível partir para o back-end e estruturar toda a parte que rodará por trás do front-end. Primeiramente, fez-se a divisão da pasta “src” em subpastas “config”, “database”, “models”, “controllers” e “routes”. Além disso, as dependências necessárias para a execução do projeto foram instaladas e o arquivo “package.json” foi criado. Nele, estão todas as informações básicas do back-end, como por exemplo o nome, a versão, as dependências, etc.

- » **Dependências utilizadas:** cors, express, http-status, sequelize, mysql2.

Após a instalação das bibliotecas, iniciou-se a estruturação do projeto através da pasta “config”. O arquivo “config.js” contém a string que realizará a conexão com o MySQL permitindo que os dados sejam manipulados no banco.

Na pasta “database”, através do arquivo “database.js”, o arquivo “config.js” é chamado e é realizada a conexão com o MySQL através da dependência “sequelize”.

Após a estruturação dos códigos que realizarão a conexão com o banco de dados, realizou-se a criação das duas tabelas que fazem parte do projeto, Usuário e Livro.

Para isso, foi criado na pasta “models” os arquivos “livro.js” e “usuário.js”. Neles as tabelas são elaboradas através do “sequelize”.

» **Atributos do livro:** id (chave-primária), nome, autor, data de lançamento, preço e ISBN.

» **Atributos do usuário:** id (chave-primária), nome, e-mail, data de nascimento e ativo.

A próxima pasta a ser trabalhada, é a pasta “controllers”. Nessa pasta, é onde acontece as requisições referentes ao CRUD (Create, Read, Update, Delete) da respectiva tabela. Para isso, foi criado um arquivo para cada tabela “livroController.js” e “usuarioController.js”.

Com o CRUD pronto, é necessário definir as rotas de acesso a cada uma das ações criadas no arquivo, bem como, os verbos HTTP (POST, GET, PUT, DELETE) referentes a cada uma. Para isso, foi criada a pasta “routes” com o arquivo “routes.js”. Abaixo estão descritas as rotas que acessam cada uma das ações definidas pelo CRUD.

```
router.post('/usuarios', UsuarioController.Insert);
router.get('/usuarios', UsuarioController.SearchAll);
router.get('/usuarios/:id', UsuarioController.SearchOne);
router.put('/usuarios/:id', UsuarioController.Update);
router.delete('/usuarios/:id', UsuarioController.Delete);
```

```
router.post('/livros', LivroController.Insert);
router.get('/livros', LivroController.SearchAll);
router.get('/livros/:id', LivroController.SearchOne);
router.put('/livros/:id', LivroController.Update);
router.delete('/livros/:id', LivroController.Delete);
```

Agora que as pastas foram estruturadas, é possível finalizar o back-end elaborando o arquivo principal “index.js”. Nesse arquivo, o servidor é criado em uma porta não utilizada pelo computador. Além disso, são definidas as requisições HTTP que serão utilizadas pelo sistema.

Para testar o back-end sem precisar ter o front-end finalizado, utilizei o software **Insomnia** para fazer requisições de POST, GET, PUT e DELETE.

» **Executando o back-end:** Para isso, é necessário acessar o terminal na pasta back-end e rodar o arquivo “index.js” através da linha de comando `<node index.js>`.

3º Parte – Front-end

Após garantir o funcionamento do back-end, iniciou-se a construção do front-end. Após criar o projeto através da linha de comando `<npx create-react-app frontend>`, as pastas foram estruturadas de modo a deixar o ambiente mais organizado. A pasta “src” foi dividida em “components”, “pages” e os arquivos principais. Na pasta “components”, foi criada a pasta “Header” para realizar o cabeçalho da página principal.

No diretório “pages”, foi criado duas pastas que se referem as páginas de cada uma das entidades, Livros e Usuários. Cada uma dessas pastas conterá a realização das páginas web referentes aos dois itens descritos. Será uma página para cada ação do CRUD (Create, Read, Update e Delete), além disso, terá uma página onde é possível visualizar todos os cadastros que foram realizados.

Abaixo encontram-se todas as páginas realizadas.

» **Página: HOME.**



» **Página:** Cadastro de usuário.

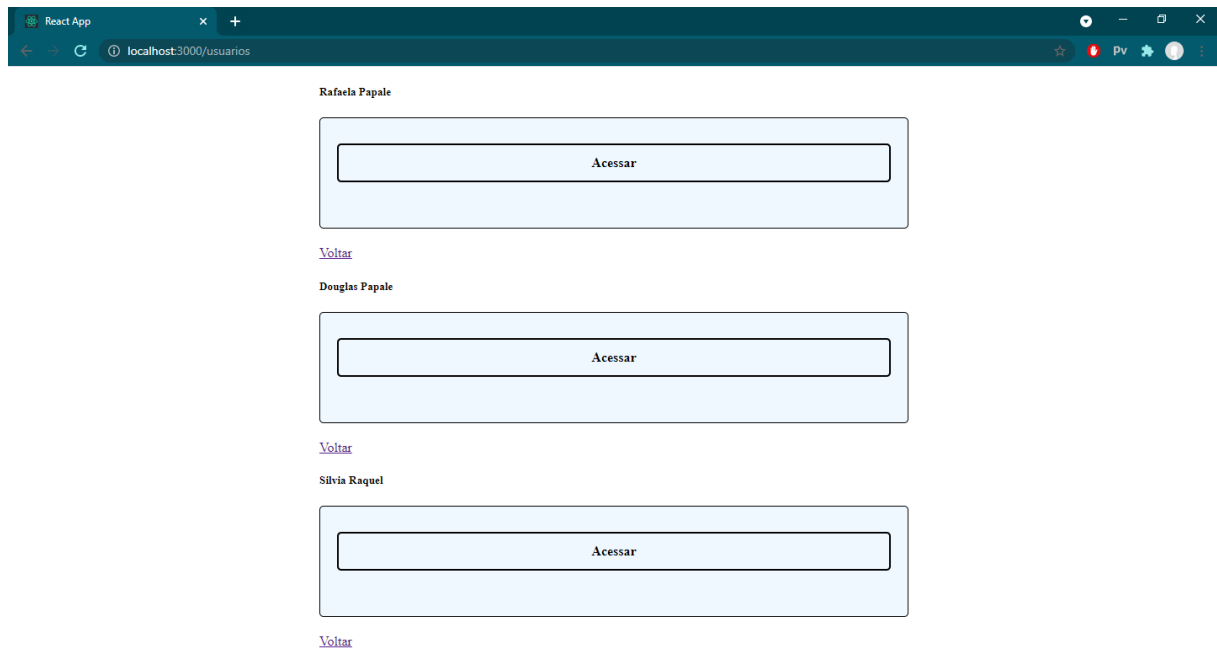


The screenshot shows a web browser window with the title 'React App' and the address bar displaying 'localhost:3000/criarUsuario'. The page content is a registration form titled 'Criar Usuário'. It contains the following fields and elements:

- Nome:** A text input field with the placeholder text 'Nome'.
- Email:** A text input field with the placeholder text 'Email'.
- Data de Nascimento:** A date picker field showing 'dd/mm/aaaa'.
- Ativo:** A radio button that is selected.
- Inativo:** A radio button that is not selected.
- Cadastrar:** A button to submit the form.

Below the form, there is a link labeled 'Voltar'.

» **Página:** Menu de usuários.



The screenshot shows a web browser window with the title 'React App' and the address bar displaying 'localhost:3000/usuarios'. The page content is a user menu titled 'Menu de usuários'. It displays a list of users, each with a name and an 'Acessar' button:

- Rafaela Papale:** A light blue box containing an 'Acessar' button.
- Douglas Papale:** A light blue box containing an 'Acessar' button.
- Silvia Raquel:** A light blue box containing an 'Acessar' button.

Below each user entry, there is a link labeled 'Voltar'.

» **Página:** Visualizar informações de um usuário.



» **Página:** Atualizar informações de um usuário.



» **Página:** Deletar usuário.



Deletar Usuário

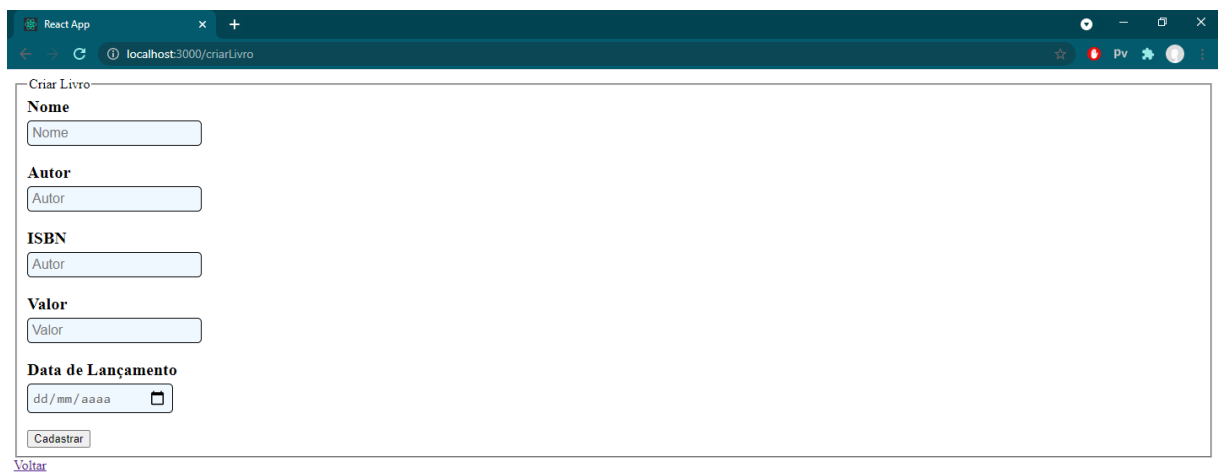
Rafaela Papale

Tem certeza que deseja deletar este registro?

[Remover](#)

[Voltar](#)

» **Página:** Cadastro de livros.



Criar Livro


Nome

Autor

ISBN

Valor

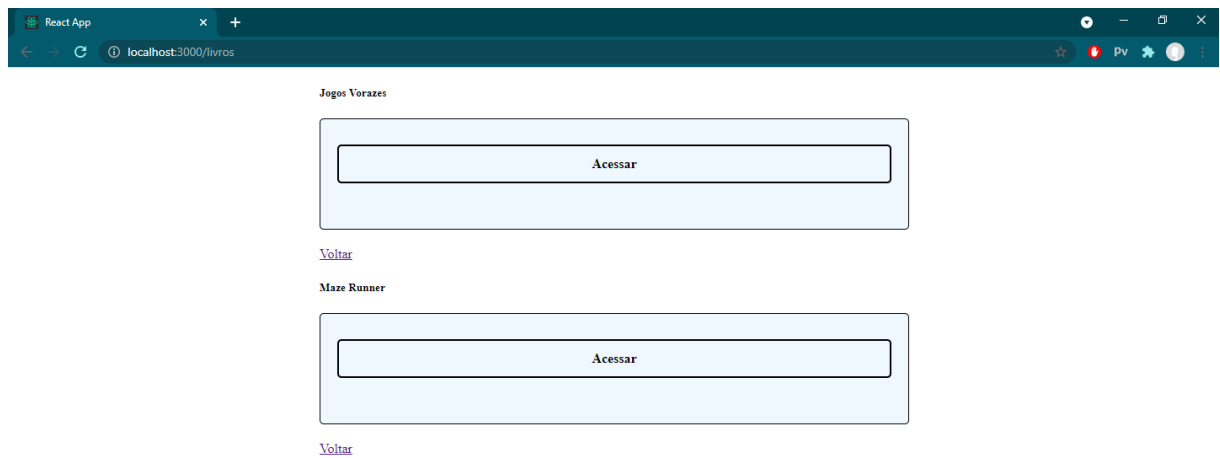
Data de Lançamento



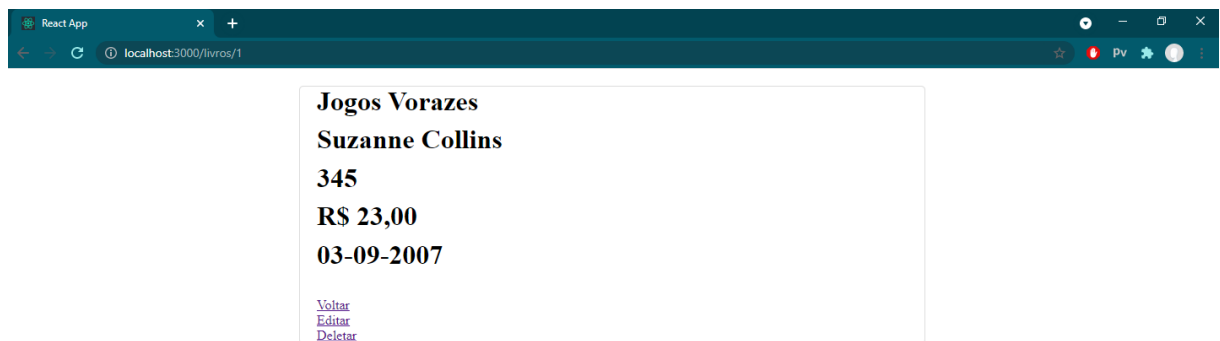
[Cadastrar](#)

[Voltar](#)

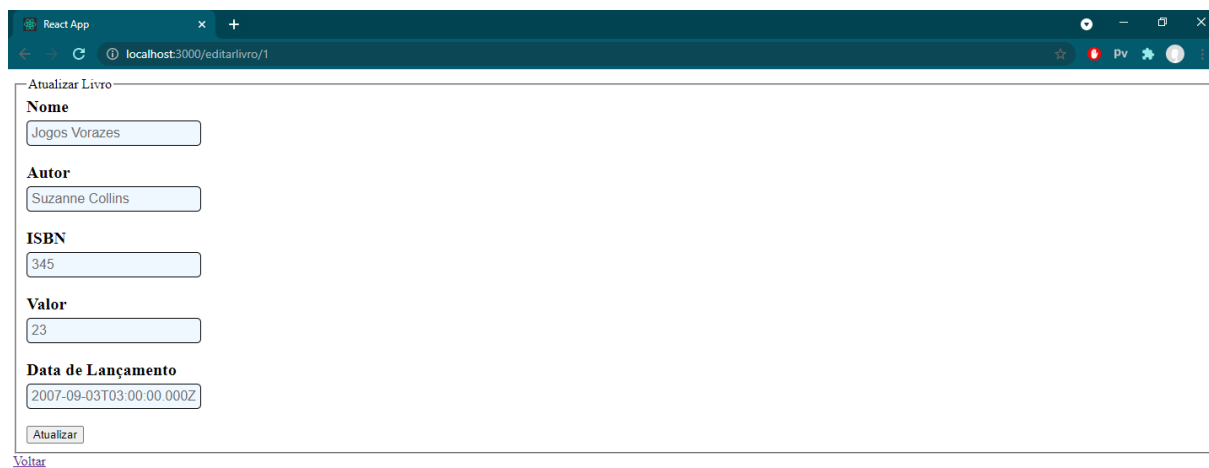
» **Página:** Menu de livros.



» **Página:** Visualizar informações de um livro.



» **Página:** Atualizar informações de um livro.



React App

localhost:3000/editarlivro/1

Atualizar Livro

Nome

Jogos Vorazes

Autor

Suzanne Collins

ISBN

345

Valor

23

Data de Lançamento

2007-09-03T03:00:00.000Z

Atualizar

[Voltar](#)

» **Página:** Deletar livro.



React App

localhost:3000/deletarlivro/1

Deletar Usuário

Jogos Vorazes

Tem certeza que deseja deletar este registro?

Remover

[Voltar](#)

No arquivo “App.js”, contido na pasta “src” são definidas todas as rotas que acessarão cada uma das páginas demonstradas acima.


```
<Route exact path="/" component={Home} />
```

```
<Route exact path="/usuarios" component={MainUsuario} />  
<Route path="/usuarios/:id" component={DetalhesUsuario} />  
<Route path="/criarUsuario" component={CriarUsuario} />  
<Route path="/editarUsuario/:id" component={EditarUsuario} />  
<Route path="/deletarUsuario/:id" component={DeletarUsuario} />
```

```
<Route exact path = "/livros" component={MainLivro} />  
<Route path = "/livros/:id" component={DetalhesLivro} />  
<Route path = "/criarLivro" component={CriarLivro} />  
<Route path = "/editarLivro/:id" component={EditarLivro} />  
<Route path = "/deletarLivro/:id" component={DeletarLivro} />
```

» **Executando o front-end:** Para isso, é necessário acessar o terminal na pasta front-end e rodar através da linha de comando **<npm start>**.

4º Parte – Pendências

1 – Não houve tempo hábil para implementar a relação entre Usuário e Livro. Dessa forma, não foi realizado o sistema de empréstimo e devolução de livros entre os usuários;

2 - O tempo foi insuficiente para realizar uma interface mais amigável e que atenda de forma adequada a solução;

3 – Realizar autenticação do Usuário;

4 – Requisito obrigatório: REST.